

Day 3. Todos: 第一个动态页面应用

作者：王琦

创建日期：2017 年 5 月 8 日

最后修改：2017 年 5 月 9 日

还记得昨天你实现的静态 Todo 页面吗？每当浏览器刷新时，这个页面的内容都将被重新清空。今天，你将借助 JavaScript 的后端框架 `express` 和 `mysql` 数据库，将原本的 Todo 改造为一个动态应用，使得每次刷新时，原本的内容得以保留。

1. 配置环境

下载框架和第三方库：

```
cnpm install --save-dev express // 下载 express
cnpm install --save-dev mysql // 下载用于连接数据库的 mysql.js
```

2. 参考代码

`day3/workspace` 基于 `express` 和 `mysql`，实现了一个简单的用户注册和登陆系统。

请阅读 `day3/workspace` 目录下相关代码，理解：

- 从 **前端页面** 向 **后端服务器** 发起请求的方式
 - HTML/JavaScript 如何发起 get 或 post 等请求
 - URL 的命名可以考虑遵循 RESTful 架构：[《理解RESTful架构》](#)、[RESTful API 设计指南](#)
 - express 后端如何收到请求
- mysql 的读写方法

3. 开发第一个动态页面应用

3.1 任务

借助 JavaScript 的后端框架 `express` 和数据库，将原本的 *Todo* 改造为一个动态应用，使得每次刷新时，原本的内容得以保留。这个 Todo List 应用应包含的功能有：

- 添加一个 `活跃的(active)` item (类似的，可以用回车键表示确认；或者增添一个添加按钮)
- 将某个 item 标记为 `活跃的(active)` 或 `完成的(completed)`

- 查看某种类别的 item
 - 查看 活跃的(active) items
 - 查看 所有的(all) items
 - 查看 完成的(completed) items
- 删除 item
 - 删除某个 item
 - 批量删除所有已完成的 items

3.2 工作流程

1) 在项目主页上记录分工

在浏览器内打开项目主页 `xx.xx.xx.xx/gitlab/your_group_name/your_project_name` , 在 `issues` 页面记录大家的工作内容以及应完成的时间（建议精确到小时）

2) 进行开发

基于 dev 分支创建新的分支（建议分支名与你所负责的功能相关），进行开发。

3) 发起并处理“合并请求” (Merge Request)

在完成开发后，将代码 push 到远程。之后，在项目主页的 `Merge Requests` 上向你的队友 `B` 发起一个由 `当前分支` 向 `dev` 分支的合并请求。`B` 在完成代码审查后，可以批注应该修改的内容，建议你重新修改；或者接受请求，合并这份代码。